Basics of TSAR Batch Control Files (bcf's) 3/20/2000

Batch Control files (bcf's) are used in TSAR to set up processing calls that will be done on a regular basis. BCF's are SAS scripts which call TSAR macros to perform tasks in sequence. It can be run once or multiple times during a processing cycle for a survey. Typically the BCF includes calls to update series, perform math between series, run X-12, and maintain the series span dates on the descriptor file. Each task is called by invoking a SAS macro routine stored in /tsar-soft/programs. Many routines require an ASCII parameter files which provides the directives to the task to be performed. Each macro has specific requirements for the setup of the parameter file, but there are similarities between them.

Typically the batch control files are written and maintained by the programming branch responsible for the production processing for that survey, but for some surveys, such as M3, survey analysts may take responsibility themselves. They are placed in the control directory for the survey and are named such that the names end in .bcf. Many surveys have one main bcf that is run and have the same name as the survey itself followed by .bcf.

The following items are required in all bcfs:

1) a macro variable named SURVEY must be defined. This is what tells TSAR which survey is being processed, and which directories to use to read/write data.

e.g. %let survey=bps;

2) A libname must be established for the series directory for the survey. Typically this is done in the programs directory for a particular survey and then %include'd in the bcf.

e.g. libname bps '/tsar-data/bps/series';

3) An include of /tsar-soft/programs/help_mac.sas should be %include'd. This calls in smaller utility macros that are called by the main routines.

4) Each of the TSAR routines that is needed is %include'd into the bcf so that it can be found when it is called. Below is a list of the macro names that can be invoked in a bcf and the filename that must be included in order for the call to be successful.

%bmark	/tsar-soft/programs/drv_bmark.sas
%clr_work	/tsar-soft/programs/clr_work.sas
%drv_grph	/tsar-soft/programs/drv_grph.sas
%drv_prnt	/tsar-soft/programs/drv_prnt.sas
%drv_rake	/tsar-soft/programs/drv_rake.sas
%drv_summ	/tsar-soft/programs/drv_summ.sas
%math	/tsar-soft/programs/math.sas
%pr_bea	/tsar-soft/programs/print_bea.sas
%pr_trann	/tsar-soft/programs/print_trann.sas
%pr_trans	/tsar-soft/programs/print_trans.sas
%pr_trans1	/tsar-soft/programs/print_trans1.sas
%pr_trans2	/tsar-soft/programs/print_trans2.sas
%seas	/tsar-soft/programs/drv_x12.sas

%trend	/tsar-soft/programs/trend.sas
%update	/tsar-soft/programs/update.sas
%x12	/tsar-soft/programs/run_x12.sas
%zero	/tsar-soft/programs/zero_out.sas

The remainder of the bcf are the macro calls themselves. Since a bcf is a SAS program, the user may insert any additional SAS code that is related to the TSAR work. Most surveys read the stat period being processed as it is used in the names of parameter files. Also, It is not unusual to find call imbedded in if-then-else blocks, or for %sysexec calls to be found to invoke UNIX system calls.

An example bcf is attachment A which includes explanatory footnotes which are explained in attachment B.

```
/* Batch Control File for Building Permits Survey  */

options linesize=72 pagesize=55 nonotes fullstimer ;
*
/* For batch processing purposes only! */
%let survey=bps ;                          (Set the survey variable)
data _null_;                               (get the stat period)
  call symput('datayear',substr(sysparm(),1,4));
  call symput('curmonth',substr(sysparm(),5,2));
  call symput('phase',substr(sysparm(),7,1));
run;

%include '/tsar-data/bps/programs/bps_libs.sas' ;     (includes: libname bps '/tsar-data/bps/series';) [1]
%include '/tsar-soft/programs/help_mac.sas' ;         (This and following %includes bring in TSAR macros to execute)
%include '/tsar-soft/programs/update.sas' ;
%include '/tsar-soft/programs/math.sas' ;
%include '/tsar-soft/programs/drv_rake.sas' ;
%include '/tsar-soft/programs/clr_work.mac' ;
%include '/tsar-soft/programs/run_x12.sas' ;

/* START RUNNING! */
%update("/tsar-data/bps/control/bps-updt-&datayear&curmonth-&phase");     (Call macro to update series with new data)
[2]
%put Step bps-updt-1-&datayear&curmonth completed.;                       (Writes message of completion to log)
%clr_work;                                                                (Clears work library) [3]
%math("/tsar-data/bps/control/bps-math-16-ut");                           (Call macro to perform series math) [4]
%put Step bps-math-16-ut completed.;
%clr_work;
%math("/tsar-data/bps/control/bps-math-sum-ut");
%put Step bps-math-sum-ut completed.;
%clr_work;
%math("/tsar-data/bps/control/bps-math-11");
%put Step bps-math-11 completed.;
%clr_work;
%sysexec(rm /tsar-data/bps/x12/x12output/x*.d16);                         (Invoke a UNIX command) [5]
%put UNIX system return code is &sysrc {removed D16 files};
%x12("/tsar-data/bps/control/bps-seas-11");                               (Call X-12) [6]
%put Step bps-seas-11 completed.;
%clr_work;
%math("/tsar-data/bps/control/bps-math-unraked");                         (More math) [7]
%put Step bps-math-unraked completed.;
%clr_work;
%math("/tsar-data/bps/control/bps-math-us1");
%put Step bps-math-us1 completed.;
%clr_work;
%drv_rake("/tsar-data/bps/control/bps-rake-us");                          (Perform raking) [8]
%put Step bps-rake-us completed.;
%clr_work;
%math("/tsar-data/bps/control/bps-math-2+");                              (And more math)
%put Step bps-math-2+ completed.;
%clr_work;
%math("/tsar-data/bps/control/bps-math-round");
```

```
%put Step bps-math-round completed.;
%clr_work;
%math("/tsar-data/bps/control/bps-math-sum");
%put Step bps-math-sum completed.;
%clr_work;

/* The following statement was inserted so that the END date is updated in the descriptor files */
%des_date(2);                                          (Update series end-dates in descriptor file)
%clr_work;

/* The following scripts were added so that the proper series are copied to their respective datasave phase series. */

%macro ifblock;                                        (User code to control what done when) [10]
 %if &phase = 1 %then %do;     * Preliminary;
  %math("/tsar-data/bps/control/bps-copy-prelim");
  %put Step bps-copy-prelim completed.;
  %clr_work;
 %end;

 %if &phase = 2 %then %do;     * Revised;
  %math("/tsar-data/bps/control/bps-copy-revised");
  %put Step bps-copy-revised completed.;
  %clr_work;
 %end;

 %if &phase = 3 %then %do;     * Final;
  %math("/tsar-data/bps/control/bps-copy-final");
  %put Step bps-copy-final completed.;
  %clr_work;
 %end;
%mend;

%ifblock;

%sysexec(nohup /tsar-data/bps/x12/x12output/x12printscr);          (Invoke user-written script) [11]
%put UNIX system return code is &sysrc {ran x12 script file};

%sysexec(mv /tsar-data/bps/x12/x12output/bpsx12  /tsar-data/bps/x12/x12output/x12out-&datayear&curmonth-&phase);
%put UNIX system return code is &sysrc {renamed X12 output};

%sysexec(lpr -Pecp-2113-4b /tsar-data/bps/x12/x12output/x12out-&datayear&curmonth-&phase);
%put UNIX system return code is &sysrc {printed X12 output};

options notes;  * To get the fullstimer notes! ;
```

1) This defines the lib name for the series directory to correspond with the survey macro variable. In this case the survey is bps so the libname statement looks like: LIBNAME BPS '/tsar-data/bps/series'; This is stored in a file which is included into the BCF, but this statement could also have been hard coded.

2) The update macro modifies existing data in a series, or adds new data points to the series. The parameter file (in this case named /tsar-data/bps/control/bps-updt-&datayear&curmonth-&phase) contains directives to the update macro to indicate what series to update, the data point to add or modify, and the value itself. An example parameter file for the update macro is as follows:

```
bps
freq 12
utnm1umu    200002 6069              V
utmm1umu    200002 12892             V
utsm1umu    200002 36970             V
utwm1umu    200002 18258             V
utnn1umu    200002 653               V
utmn1umu    200002 2048              V
utsn1umu    200002 6109              V
utwn1umu    200002 2368              V
```

The first line in this parameter file is survey, and the second line is frequency (12 is monthly, i.e. 12 data points in a year). The remaining lines are the specifications for the data to update.

3) The macro clr_work is defined as follows:

```
%macro clr_work;
proc datasets library=work kill nolist;
run;
quit;
%mend;
```

and simply deletes any datasets that exist in the SAS work library.

4) The macro math manipulates the data series. The term "math" is a little misleading. The math macro can perform any of the things users consider to as math, such as addition, multiplication, or rounding, but can also invoke many other SAS functions. In fact the directives in the math parameter files are SAS statements that are brought into the math macro and then executed. An example math parameter file could look as follows:

```
SPAN  1  END  END
UTNETOMU = UTNE1UMU + UTNE2UMU + UTNE34MU + UTNE5PMU;
UTMWTOMU = UTMW1UMU + UTMW2UMU + UTMW34MU + UTMW5PMU;
UTSOTOMU = UTSO1UMU + UTSO2UMU + UTSO34MU + UTSO5PMU;
UTWETOMU = UTWE1UMU + UTWE2UMU + UTWE34MU + UTWE5PMU;
XTNE1UMU = ROUND(UTNE1UMU/(FTNE1UMU/100));
XTMW1UMU = ROUND(UTMW1UMU/(FTMW1UMU/100));
XTSO1UMU = ROUND(UTSO1UMU/(FTSO1UMU/100));
XTWE1UMU = ROUND(UTWE1UMU/(FTWE1UMU/100));
```

The first line of a math directives file is always a SPAN record. The number following the word SPAN is the number of spans specified. There can be up to three. The span indicates which data points to start and stop with. In this example, only the last data point in the series' will be used for the calculations. The span

directives can contain START, END, or additions to these. Other valid span points are: START END to perform math on the entire series or END-3 END, which would perform math on only the last four data points in the series.

5) Users may perform UNIX commands through the use of %sysexec which is a built-in SAS macro. In this case, the user is deleting the D16 output tables prior to running X-12. This ensures that if X-12 doesn't properly create a new table, a prior existing table will not be used in its place.

6) X-12 is called by invoking the macro named X12. A sample parameter file for the X-12 directives could look like:

```
XTNE1UMU  1    d16=STNE1UMU
XTMW1UMU  1    d16=STMW1UMU
XTSO1UMU  1    d16=STSO1UMU
XTWE1UMU  1    d16=STWE1UMU
XTUS24MU  1    d16=STUS24MU
XTUS5PMU  1    d16=STUS5PMU
```

This parameter file would indicate the there are six series which require seasonal adjustment. The output from the D16 table is then input into the series name following the =.

7) There is no set order to calling the macros, and the bcf needs to be set up so that things are done in proper sequence. In this example, series math on the output from X-12 cannot be done until the X-12 work is completed. Another observation on this example bcf, is that there are back-to-back calls to the math macro. It made more sense to the user to lump together the related math directives into multiple files, but they could have all been in the same file. The directives are executed in the order in which they appear on the directives file. Also some directives may apply to a different span of data points so would require a separate parameter file.

8) The rake macro will force addition of component series to another series. An example parameter file to the rake macro is as follows:

```
SPAN  1  END  END
RAKE(DTUSTOMU,DTNETOMU,DTMWTOMU,DTSOTOMU,DTWETOMU);
```

In this example, the four series dtnetomu, dtmwtomu, dtsotomu, and dtwetomu are forced to add to dtustomu (i.e. four regions should sum to US total). Because the span is END END, then only the last data point of dtustomu is updated.

9) The des_date macro will update the descriptor file with the starting and/or ending points based upon what the series actually contains. In other words, it inputs each series in the series directory, and checks the actual stat period for the beginning/ending data points and then updates the appropriate record in the descriptor file. The des_date macro takes one paramater: '1' will update only the beginning dates, '2' will update only the end dates, and '0' will update both. In this example, only the end dates will be updated. Unless new series are created, it is very rare that data is added to the beginning of a series. So generally only the '2' parameter is used in a bcf.

10) Since the bcf is simply SAS macro code, the user can control what is done under different conditions. In this example, the user is maintaining separate series for "preliminary"," revised", and "final" data. One parameter that is input to the bcf indicates which of these three "phases" of updates is being processed. In this example, a macro if block conducts the test and controls which of three possible math files is executed.

11) Just as the user can invoke a UNIX command through the use of %sysexec (see #5 above), a user-written ksh script can also be executed.